IN THE CLAIMS

Please cancel claims 1 and 5 without prejudice or disclaimer.

Please replace all prior versions, and listings, of claims in the application with the following list of claims:

1. (Canceled)

2. (Previously Presented) A method of debugging a target system connected to a host computer, the target having a digital signal processor with a memory including a reserved storage location designate as a vector, said memory further storing an application program, the method comprising:

loading a stack into said memory;

storing in said reserved location information indicative of said stack location;

dynamically loading a computer file into said memory, said file containing a subroutine required for use by said application program; and

storing at a predetermined location in said stack data indicative of an entry point into said dynamically loaded file;

running said application on said target, whereby said application accesses said vector to thereby call said entry point and thus run said subroutine.

3. (Original) A method as claimed in claim 2 wherein the subroutine allows transfer of data to or from the host computer.

4. (Original) A method as claimed in claim 2 wherein the subroutine allows said application to work round a hardware error in said target.

5. (Canceled)

6. (Previously Presented) The method of claim 2 wherein said predetermined one of said addressable locations is a predetermined address in the stack of an exception handler.

7. (Original) A debugging device comprising a target system connected to a host computer, the target having a digital signal processor with a memory including a reserved storage location designated as a vector, said memory further storing an application program, the device comprising:

first loading circuitry for loading a stack into said memory;

vector writing circuitry for storing in said reserved location information indicative of said stack location;

dynamic loading circuitry in said host for loading a computer file into said memory of said digital signal processor, said file containing a subroutine required for use by said application program;

stack writing circuitry for storing at a predetermined location in said stack data indicative of an entry point into said dynamically loaded file;

wherein when said digital signal processor runs said application, said application accesses said vector to thereby call said entry point and thus runs said subroutine.

8. (Original) The device of claim 7 wherein the subroutine allows transfer of data to or from the host computer.

9. (Original) The device of claim 7 wherein the subroutine allows said application to work round a hardware error in said target.

10. (Previously Presented) A device for operating an embedded digital signal processor, said embedded signal processor having a memory comprising plural addressable locations, and being adapted to run an application,

the device comprising a host computer connected to said embedded digital signal processor, said host computer comprising a computer file including a subroutine required for said application;

said host computer comprising a linker-loader connected to said link and operative to send said file and dynamically load said file to said memory of said embedded signal processor whereby said file has an entry point at one of said addressable locations, said linker-loader comprising means for storing at a predetermined one of said addressable locations data representative of the address of said entry point, wherein said predetermined one of said addressable locations is a predetermined address in the stack of an exception handler;

said embedded digital signal processor comprising processor circuitry running said application whereby said application determines said data representative of said address, thereby accessing said file to enable said application to run.